

"Express Mail" mailing label number EV 327 682 819US

Date of Deposit: March 17, 2004

ELECTRONIC DEVICE NETWORK PROVIDING STREAMING UPDATES

ATTORNEY DOCKET NO.: 14908US02

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit of priority of U.S. Provisional patent application having serial number 60/455,184, filed on March 17, 2003, and hereby incorporates herein by reference the complete subject matter thereof, in its entirety.

[0002] The present application also hereby incorporates herein by reference the complete subject matter of PCT application having publication number WO/02/41147 A1, and having application number PCT/US01/44034, filed on November 19, 2001, in its entirety.

[0003] The present application also hereby incorporates herein by reference the complete subject matter of U.S. Provisional patent application having serial number 60/249,606, filed on November 17, 2000, in its entirety.

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0004] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[0005] [Not Applicable]

BACKGROUND OF THE INVENTION

[0006] Electronic devices, such as mobile phones and personal digital assistants (PDA's), often contain firmware and application software that are either provided by the manufacturers of the electronic devices, by telecommunication carriers, or by third

parties. These firmware and application software often contain software bugs. New versions of the firmware and software are periodically released to fix the bugs or to introduce new features, or both.

[0007] Electronic devices, such as mobile handsets, access servers to retrieve updates that are needed to update firmware and/or software. Creating efficient and compact updates for firmware/software updates is important. Creating an update agent that can process these update packages efficiently on the mobile handset is also a great challenge. Creating such solutions adapted to work in a plurality wireless networks is also important.

[0008] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of skill in the art, through comparison of such systems with some aspects of the present invention as set forth in the remainder of the present application with reference to the drawings appended hereto.

SUMMARY OF THE INVENTION

[0009] Aspects of the present invention may be found in a stream processing generator of difference information. The stream processing generator may comprise a first stream of information. The first stream of information may comprise a plurality of first bytes of data. The stream processing generator may also comprise a second stream of information. The second stream of information may comprise a plurality of second bytes of data. The stream processing generator may also comprise an operation array storing operations for tree-based encoding of the first and second streams of information. The stream processing generator may simultaneously transform the first and second streams of information, analyze the plurality of first and second bytes of data encountered in the first and second streams of information, determine difference information between the first and second streams of information, and output the difference information between the first and second streams of information.

[0010] In an embodiment according to the present invention, the stream processing generator may further comprise a differencing component computing and expressing the difference information employing a differencing instruction set.

[0011] In an embodiment according to the present invention, the differencing instruction set may comprise at least one operation selected from a match operation, an insert operation, a delete operation, and a replace operation.

[0012] In an embodiment according to the present invention, the stream processing generator may further comprise an encoder providing tree-based encoding. The encoder may employ a block-based hierarchical representation. The encoder may also segment blocks during encoding.

[0013] In an embodiment according to the present invention, the encoder may also employ variable length encoding techniques for operators in a set of operations. The encoder may also employ tree-based variable sized blocks. The stream processing generator may compute a cumulative address offset.

[0014] Aspects of the present invention may be found in a wireless electronic device network adapted to dispense streaming updates to at least one of a plurality of electronic devices. The streaming updates may be for updating at least one of firmware and software. The wireless electronic device network may comprise a stream processing generator generating streaming updates. The stream processing generator may employ stream processing to at least one of a plurality of blocks of content. The at least one of a plurality of blocks of content may comprise a stream of bytes. The stream processing generator may process the at least one of a plurality of blocks of content until reaching an end of the stream of bytes. The wireless electronic device network may also comprise an electronic device server communicatively coupled to the at least one of a plurality of electronic devices. The electronic device server may disseminate the streaming updates to the at least one of a plurality of electronic devices. The wireless electronic device network may also comprise a streaming update processor in the at least one of a plurality of electronic devices for processing the streaming updates received from the electronic device server.

[0015] In an embodiment according to the present invention, the stream processing generator may employ a transform array to store operations needed to transform a first stream of information into a second stream of information. The stream processing generator may process the first stream and the second stream at a byte-by-byte level to generate streaming updates. Each byte may be one of a text character and a binary character of at least one of the first and second streams of information.

[0016] In an embodiment according to the present invention, the stream processing generator may maintain a minimum weight transform array wherein a minimum weight may be assigned to a set of operations. The minimum weight may be computed by employing an edit distance computation in management of an operational array.

[0017] In an embodiment according to the present invention, the set of operations may comprise at least one of a replace operation, a match operation, an insert operation, and a delete operation.

[0018] In an embodiment according to the present invention, the stream processing generator may comprise an encoder for encoding operations. The encoder may assign values to each operation in a set of operations.

[0019] In an embodiment according to the present invention, the encoder may assign a minimum weight. The minimum weight may be computed by employing appropriate weights in management of a transform array.

[0020] In an embodiment according to the present invention, the stream processing generator may comprise an encoder for encoding operations. The encoder may assign numeric values to the operations. Non-zero values may be assigned to replace operators and insert operators and zero may be assigned to match operators.

[0021] In an embodiment according to the present invention, the encoder may assign a minimum weight. The minimum weight may be computed by employing appropriate weights in management of a transform array.

[0022] In an embodiment according to the present invention, the streaming updates may comprise a difference output for two streams comprising one of binary data and text data. The difference output may comprise at least one of a tree map, operational codes for operations comprising at least one of a replace operation, a match operation, a delete operation, an insert operation, and data characters associated with at least one of the insert operation and the replace operation.

[0023] In an embodiment according to the present invention, a first stream of information and a second stream of information may be processed by the stream processing generator. The first and second streams of information may each be file-based streams. An operational array may be computed in the stream processing generator by consuming each of the file-based streams in small chunks. A small chunk may comprise one of a 64-byte block of information, a 16-byte block of information, and a 4-byte block of information.

[0024] In an embodiment according to the present invention, each small chunk of each of the file-based streams may be used to define a portion of an operational array. After encoding each operation in an encoder, the encoder may reorient to a corresponding point in each of the file-based streams to start additional encoding of a next small chunk.

[0025] In an embodiment according to the present invention, the electronic device may comprise at least one of a plurality of mobile electronic devices, and wherein the plurality of mobile electronic devices comprise at least one of a mobile cellular phone handset, personal digital assistant, pager, MP3 player, and a digital camera.

[0026] Aspects of the present invention may be found in a method of generating streaming updates by converting a first stream of information into a second stream of information for updating an electronic device. The method may comprise identifying the first and second streams of information, accessing the first and second streams of information, retrieving one block of content at a time from each of the first and the second streams of information, determining a transform operation, executing the transform operation, computing an output from the transform operation, creating a tree-based transform output from operators determined in the transform, encoding the tree-based transform output employing at least one of variable length encoding and fixed length encoding, outputting difference information into at least one memory structure, determining whether additional blocks of content are to be processed by evaluating the first and second streams of information, retrieving an additional block of content from each of the first and the second streams of information upon determining that additional blocks of content are to be encoded, continuing encoding until reaching an end of a stream of blocks of content to be encoded, compressing difference information output, and packaging the difference information output into an update.

[0027] In an embodiment according to the present invention, the method may further comprise buffering content from the first stream of information and the second stream information to determine the difference information and encoding the difference information before outputting the difference information.

[0028] In an embodiment according to the present invention, the update may facilitate conversion of the first stream of information into the second stream of information. Retrieving blocks of content from the second stream of information may be performed at a fixed pace using a fixed block size. Retrieving blocks of content from the first stream of information may performed at a variable pace using a variable block size. A reference to the second stream of information may be maintained and a cumulative offset may be computed.

[0029] In an embodiment according to the present invention, a look-ahead operation may be executed as part of retrieving blocks of content. The look-ahead operation may employ data from the first and second streams of information to compute an operational array.

[0030] In an embodiment according to the present invention, a longest common sub-string technique may be employed prior to determining a transform operation.

[0031] In an embodiment according to the present invention, illegal and unexpected operation codes may be employed as an escape sequence during encoding.

[0032] In an embodiment according to the present invention, a combination of the escape sequence, a type field of two bits, and a length representing a repetition of data associated with the type field is employed by to encode long strings of complete matches between the first and second streams of information.

[0033] In an embodiment according to the present invention, the electronic device may comprise at least one of a plurality of mobile electronic devices, and wherein the plurality of mobile electronic devices may comprise at least one of a mobile cellular phone handset, personal digital assistant, pager, MP3 player, and a digital camera.

[0034] These and various other advantages and features of novelty which may characterize the invention are pointed out with particularity in the claims annexed hereto and that form a part hereof. However, for a better understanding of the invention, its

advantages, and the objects obtained by its use, reference should be made to the drawings which form a further part hereof, and to accompanying descriptive matter, in which there are illustrated and described specific examples of an apparatus in accordance with the invention.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0035] **Figure 1** is a block diagram illustrating an electronic device network adapted to dispense streaming updates to electronic devices having updatable firmware and/or software according to an embodiment of the present invention;

[0036] **Figure 1A** is a diagram illustrating a three level hierarchy employable for encoding of a block operations according to an embodiment of the present invention;

[0037] **Figure 2** is a block diagram illustrating software layers and components in an electronic device according to an embodiment of the present invention; and

[0038] **Figure 3** is a flow chart illustrating a method for providing streaming updates according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0039] Electronic devices may be adapted to access servers to retrieve updates for updating at least one of firmware and software. An electronic device may be, for example, a mobile electronic device having software/firmware, such as, mobile cellular phone handsets, personal digital assistants (PDA's), pagers, multimedia devices, such as MP3 players, DVD players, CD players, still and video cameras, etc. An update may comprise firmware and/or software updates that modify or change the version of a particular firmware or software installed in the electronic device, for example, upgrading to a newer version, repairing a bug in the firmware/software, etc. An update may also add new services to the electronic device or delete services, as desired by a service provider, device manufacturer, or an end-user.

[0040] An electronic device requesting access to a server may transmit an access request to the network using a first type of communication and the network may respond using the same type of communication. Alternatively, the electronic device may transmit an access request to the network using a first type of communication and the network may respond using a different type of communication. For example, the electronic device may transmit a request to the network using a short message service (SMS) and the network may respond to the request using a circuit switched or packet switched communication service, or the reverse may occur.

[0041] **Figure 1** is a block diagram illustrating an electronic device network adapted to dispense streaming updates to electronic devices having updatable firmware and/or software according to an embodiment of the present invention.

[0042] **Figure 1** illustrates an electronic device network 105 adapted to dispense streaming updates to electronic devices, for example, mobile handset 107. The electronic devices, for example, mobile handset 107, may be adapted to update firmware and/or software contained therein. The streaming updates may be generated by a streaming

update generator 131 and stored in an update store 129 for retrieval by a device server 127 communicatively coupled to mobile handset 107.

[0043] The mobile handset 107 may also comprise a non-volatile memory 109 and random access memory (RAM) 125. The non-volatile memory 109 may also comprise an update 123, applications 121, firmware 117, an operating system (OS) 119, a streaming update control 115, a streaming update processor 113, and an update application (UA) loader 111.

[0044] The streaming update generator 131 may employ a technique of stream-based processing wherein blocks of content/data are continuously processed from the beginning of the stream until the end of the stream is encountered.

[0045] The streaming update generator 131 may employ an array, (that may be defined as a transform array), to store operators used to transform an original file into a streaming file. The streaming update generator 131 may process streams at the byte-by-byte level, wherein each byte may be a text character or a binary value. The streaming update generator 131 may also maintain a minimum-weight transform array wherein a weight is assigned to the operators. For example, the streaming update generator 131 may employ four operators, such as for example, a replace operator (R), a match operator (M), an insert operator (I), and a delete operator (D).

[0046] During an encoding process, values may be assigned to the operators and the values assigned may be numbers, for example. Non-zero numeric values may be assigned to operator R and operator I. Zero may be assigned to operator M. A minimum weight may be computed by employing weights in management of the transform array. A difference output for two streams of binary data or textual content (files) may comprise a tree map comprising operator codes (M, R, I, and D) and data characters associated with operator I and operator R.

[0047] In an embodiment according to the present invention, a transform may be employed in the streaming update generator 131 to process two streams, wherein each

stream may be a file, an array, etc. The output may be a tree-based transform encoding that may be at least as long as a smaller of the two streams (files), and may also be smaller than the larger of the two streams (files). For example, if an original file is L1 in length and the new file is L2 in length, then the operational array, or op-array (populated with operators) may provide that: $\text{MAX}(L1, L2) < \text{op-array} < L1 + L2$.

[0048] In an embodiment according to the present invention, an op-array may be generated from the original stream and the new stream (file) being encoded. Additionally, a minimum weight may be computed by employing edit distance in management of the op-array. Hirschberg's algorithm or an edit-distance algorithm may be employed in the streaming update generator 131 to generate the op-array. The op-array may be processed by means of tree-based transform encoding to generate difference information that may be packaged along with other related information, such as a tree map, into an output.

[0049] In an embodiment according to the present invention of the streaming update generator 131, an op-array may be computed by employing an approach that consumes the two streams (files) in small chunks, such as in 64-byte blocks. Each of the 64-byte blocks, or other such small sections of input streams, may be used to define a portion of the op-array. After encoding a first operation, processing reorients to a corresponding point in both streams to start encoding a next operation from there.

[0050] In an embodiment according to the present invention, the four operators may be represented by variable length codes based on frequency of occurrence of the associated operations. Because M is the most frequently encountered operation, operator M may be given a value of 0. R being the second most frequently encountered operation, may have a symbol of 10, etc. The average length per symbol across the entirety of the op-array may be approximately 2.

[0051] In an embodiment according to the present invention, two bits may be used to represent each of the four operations.

[0052] In an embodiment according to the present invention, the streaming update generator 131 may employ Hirschberg's techniques to determine the op-array for each block of content in the original file. Hirschberg's algorithm is specified in CACM 18(6) 341-343 1975. Hirschberg (1975) presented a method of recovering an alignment (of a longest common sub-string technique, LCS) in $O(n^2)$ time but only in linear, $O(n)$ -space, i.e. Hirschberg taught that an optimal alignment from a distance matrix between two strings s_1 and s_2 may be recovered in $O(|s_1|*|s_2|)$ time and only $O(|s_2|)$ space using binary-recursion. If a string contains zero characters or one character, an alignment may easily be found. Otherwise, the approach is to cut string s_1 in the middle to form s_{1a} and s_{1b} and to find a corresponding place to cut string s_2 into s_{2a} and s_{2b} . It may be noted that s_{2a} and s_{2b} may have quite different lengths. The alignment problem may then be solved recursively on s_{1a} and s_{2a} and on s_{1b} and s_{2b} . This is an example of a divide and conquer paradigm.

[0053] After conclusion of a Hirschberg computation, an op-array may be prepared. The individual operations may be encoded, employing tree-based op-array encoding. A set of N symbols may be processed to build tree structured representation of the op-array.

[0054] **Figure 1A** is a diagram illustrating a three level hierarchy employable for encoding of a block operations according to an embodiment of the present invention. **Figure 1A** illustrates a three level hierarchy tree 100A that may be employed for encoding of a block of 64 operations, for example. In an embodiment according to the present invention, each node in top level 110A may encode N bytes. In an embodiment according to the present invention, each node in the second level 120A may encode $N/4$ bytes. In an embodiment according to the present invention, the nodes in the third level 130A may encode $N/16$ bytes. If all N bytes match, then the single bit 0 may be used in the encoding to indicate a complete match for an entire block.

[0055] Although only three levels (110A, 120A, and 130A) are illustrated in the hierarchical tree 100A in **Figure 1A**, numerous additional hierarchical levels may be employed as desired. The three-level hierarchical tree 100A, as illustrated in **Figure 1A**,

has been employed for purposes of example. In a three-level hierarchical tree, for example 100A, each intermediate node, for example 110A, in the tree may be a sub-block, wherein one intermediate node, for example 110A, may represent 16 symbols or operators. The second level down 120A may represent 4 operation nodes, and at each node, 0 or 1 may be sent. At leaves of the tree, (i.e., third level down 130A), the op-codes may be set using variable length codes (VLC). A prefix may be significant, left to right, in each op-code. For example, the VLC for the op-codes may be set as follows: M: 0, R: 10, I: 110, and D: 111.

[0056] In an embodiment according to the present invention, a difference output created by the streaming update generator 131 may be packaged with three sections, each section containing a different type of data. For example, the tree map, such as 100A, may be located in one section at one end. Any new data character may be packaged in the beginning (for operators R and operators I) and the op-codes may be set in the middle as integers. Run length encoding, when used, employ integers for length. The new data section may be considered to be most compressible. The tree map section may often be least compressible.

[0057] In an embodiment according to the present invention, the streaming operation of the streaming update generator 131 may employ each block of N bytes at a time and build a tree upon it. When the tree representation is processed, the streaming update generator 131 may employ two pointers, one for each file or stream being compared. After processing each block of N bytes, for a next block, the current locations of the two pointers may be taken into consideration to compute a cumulative offset, which may also take into consideration op-array pointers of previous block(s). The two pointers may not always be at same relative positions and they may be advanced from their current pointer positions before processing a next block.

[0058] Pointer positions may be altered when incremented or decremented, such as by (+1) for operator D and (−1) for operator I. In an embodiment according to the present invention, the pointer for the original file or stream may be adjusted while the pointer for the new file or stream may always be advanced at steps of 64, for example. In

an embodiment according to the present invention, the reverse may be true, i.e. the pointer for the new file or stream may be adjusted at each step (after processing each block) while the pointer in original file or stream may always be advanced at steps of 64, for example.

[0059] Aspects of the present invention may be found in employing a variable block size instead of fixed block sizes to the streaming update generation method. Block sizes may be reduced. Employing block-by-block processing may result in propagation of error coefficients that may be identified, consolidated, and/or repaired. In an embodiment according to the present invention, consolidation of boundary artifacts may be effected by employing a look-ahead scheme. The amount of look ahead may be adjustable, and may also be determined based upon threshold values.

[0060] In an embodiment according to the present invention, an LCS technique may be used in every block to find a next match. Using a threshold of 16 bytes, a search for an LCS may be executed.

[0061] Transforms, such as LCS and Hirschberg, may be run once for all of the blocks. Re-computing a transform array, such as an op-array, may be performed after the operators identified for the blocks involved are consumed. Therefore, the transform array may be determined once for each block.

[0062] In an embodiment according to the present invention, optimization of encoding may be employed wherein special escape sequences provide for compact encoding. For example, a tree map may be analyzed to determine whether there is any change in any portion of the tree and may employ 0 at the highest level if there are no changes in any of the sub-trees, whereas a 1 would indicate a difference in some sub-tree component. When a node has four sub-nodes, encoding 10000 is not possible, therefore 10000 may be used as an escape sequence. The sequence 10000 may therefore be used as a special instruction from encoder to decoder. Using the special instruction, directives may be communicated in the output, for scenarios such as a pure match of block by block, a run length for encoding the number of blocks, a module processing approach

wherein the special instruction identifies padded areas between modules, a special escape sequence indicating a need to jump to a next module, etc.

[0063] In an embodiment according to the present invention, the streaming update generator 131 may process two input streams (files) and discover module relocation using LCS techniques. Modules may then be aligned using information derived from the LCS techniques.

[0064] **Figure 2** is a block diagram illustrating software layers and components in an electronic device according to an embodiment of the present invention. **Figure 2** illustrates a mobile handset 205 comprising firmware 207 over which an operating system having file system 209 may execute an application having multiple components 221. Additionally, a streaming update processor 213 employing a streaming update control 211 accessing and manipulating streams may be executed on top of the operating system having file system 209 to update the application having multiple components 221 employing updates having multiple streams 223. In an embodiment according to the present invention, the streaming update controller 211 may be incorporated into the streaming update processor 213, as desired.

[0065] In an embodiment according to the present invention, the application having multiple components 221 may comprise a plurality of components, such as for example, an executable program, a shared library, a configuration file, etc. The streaming update processor 213, employing the updates having multiple streams 223, may be capable of updating one or more of the components of the application having multiple components 221. The updates having multiple streams 223 may contain a plurality of difference outputs packaged together, each of the plurality of difference outputs being created for, and used to update, a corresponding one of the plurality of components in the application having multiple components 221. The streaming update control 211 may be used to access and process original files or streams in mobile handset 205 to update the files to newer versions.

[0066] **Figure 3** is a flow chart illustrating a method for providing streaming updates according to an embodiment of the present invention. **Figure 3** may also illustrate operation of a streaming update generator, such as streaming update generator 131 in **Figure 1**. Initially, the streaming update generator may identify streams to be processed. The streams (an original and a new stream for each component to be updated) may be opened or accessed (block 307). One block of content, such as a block of 64 bytes, may be accessed or retrieved from the original and the new stream (block 309). The transform operation may be executed and the output from the transform may be computed (block 311). A tree map may be created from the operators determined in a previous transform (block 313).

[0067] Encoding of the tree-based transform output may be conducted, by employing variable length encoding or fixed length encoding (block 315). The difference information may be output to appropriate structures or resources (block 317). It may be determined whether more content is to be processed from the input streams (block 319). If more content is to be processed, then control loops back to the step of retrieving blocks of content (block 309) continuing with a next block of content. Otherwise, the difference output may be compressed, packaged, and communicated to an external repository or storage memory device (block 321).

[0068] The electronic device network 105 of **Figure 1** may employ a streaming update generator 131 to generate updates and employ a corresponding streaming update processor in an electronic device, for example mobile handset 107, to update firmware and/or software. The streaming update generator 131 may use a small buffer of content for processing original and new data files. The streaming update generator may operate on a portion of an entire file scope or a portion of an entire stream of content. Data files may also be regarded as “streams” during processing. The streaming update generator may read a portion of an entire file or a portion of an the entire stream of content to process stream/file, fetch small blocks of data, and generate difference information between streams.

[0069] Although a system and method according to the present invention has been described in connection with a preferred embodiment, the invention is not intended to be limited to the specific form set forth herein, but on the contrary, is intended to cover such alternative modifications and equivalents, as can be reasonably included within the spirit and scope of the invention, as defined by this disclosure and the appended diagrams. It is intended that the scope of the invention be limited not with this detailed description, but rather by the claims appended hereto.